



# 面向对象的软件构造实践

实验六  
(6学时)

2024春



HITSZ 实验与创新实践教育中心  
Education Center of Experiments and Innovations, HITSZ

用户界面

事件处理

图形系统

数据存储  
与展示

音乐音效

网络编程

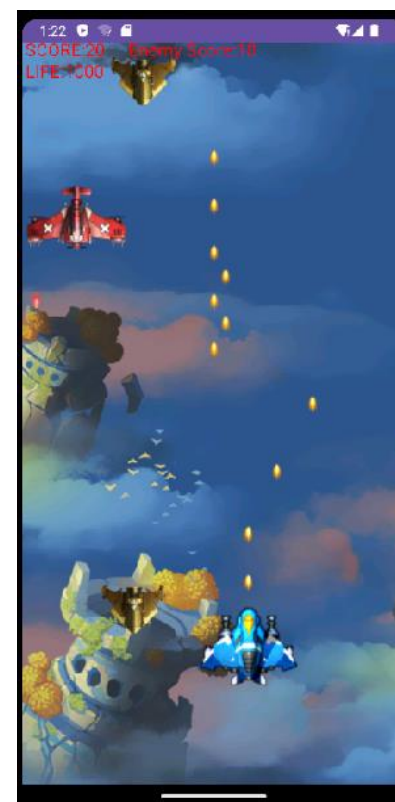
## 模块功能：通过网络传递数据

- ① 完成服务器端的开发，实现联机对战功能，在对战过程中实时显示对手分数

玩家1



玩家2



- 了解网络编程基础知识，理解客户端-服务器架构；
- 掌握Android网络通信中多线程的使用；
- 掌握基于TCP/IP的Socket网络通信的使用方式。

6.1 网络编程基本知识

6.2 Android网络通信

6.3 基于TCP/IP协议的Socket通信

## 6.1 网络编程基本知识

### IP地址

网络中的每台计算机都必须有**唯一的IP地址**作为标识，这个数通常写作一组由 “.” 号分隔的十进制数



在命令提示符中使用**ipconfig**命令可以查询本机IP地址。

```
命令提示符
Microsoft Windows [版本 10.0.22621.1992]
(c) Microsoft Corporation。保留所有权利。

C:\Users\yang>ipconfig

Windows IP 配置

无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 本地连接* 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 WLAN:

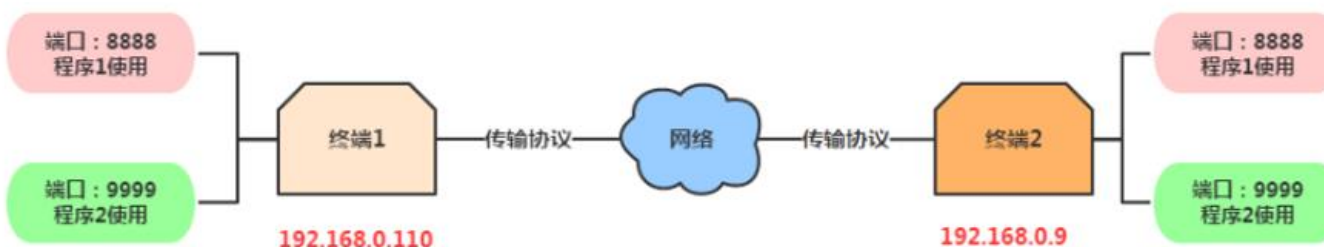
    连接特定的 DNS 后缀 . . . . . :
    IPv6 地址 . . . . . : 2001:250:3c0f:2000::7c14
    本地链接 IPv6 地址 . . . . . : fe80::6797:8ffc:da79:e984%4
    IPv4 地址 . . . . . : 10.250.102.144
    子网掩码 . . . . . : 255.255.0.0
    默认网关 . . . . . : fe80::3e15:fbff:fea9:722f%4
                        10.250.0.1

以太网适配器 以太网 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :
```

## 6.1 网络编程基本知识

### 端口



- “端口号” 用来标识同一台计算机上运行的**进程**（程序），范围是0~65535。
- 每个被发送的网络数据包也都包含有“端口号”，用于将该数据帧交给具有相同端口号的应用程序来处理。
- **套接字**: **<IP>:<端口>**

### 客户端-服务器架构



- 飞机大战APP采用C/S架构，由服务器项目提供网络服务；
- 客户端项目运行在模拟器中，服务器项目运行在本地主机，客户端连接服务器时可指定服务器IP为**10.0.2.2**。
- 如客户端运行在真机，需使用服务器主机IP发送请求。

## 6.2 Android网络通信

- 任何Android应用需要访问到网络，都要求被赋予访问网络的权限。
- 在开发网络应用时，首先要在**AndroidManifest.xml**中添加网络访问权限。一般是放在<application>节点之前。声明网络访问权限代码：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ee.example.client_multithread">
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Client_MultiThread"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter...>
        </activity>
    </application>
</manifest>
```



## 6.2 Android网络通信

- Android在主线程中进行网络请求会出现NetworkOnMainThreadException
- 开启**子线程**进行网络通信，使用**Handler**在子线程和主线程中进行通信

## 6.3 基于TCP/IP的Socket通信

### Socket通信过程

#### 服务器端编程步骤

1. 创建服务器项目；
2. 创建服务器端套接字并绑定到一个端口上；
3. 套接字设置监听模式等待连接请求；
4. 接受连接请求后进行通信；
5. 返回，等待下一个连接请求；
6. 通信结束关闭套接字。

#### 客户端编程步骤

1. 创建客户端套接字（指定服务器端IP地址与端口号）；
2. 连接（Android创建Socket时会自动连接）；
3. 与服务器端进行通信；
4. 通信结束后，关闭套接字。

## 6-1 Socket客户端

- 在游戏首页添加“联机对战”按钮，点击此按钮等待匹配对手,匹配成功则开始游戏，游戏难度可默认为任意难度，也可让玩家选择难度；
- 游戏中每隔一段时间向服务器发送自己的得分；
- 将服务器传来的对方的分数显示在游戏页面；
- 游戏结束显示对战结果。

## 6-2 Socket服务器端

- 监听玩家匹配请求，匹配玩家；
- 接收一方玩家分数，发送给对方玩家。

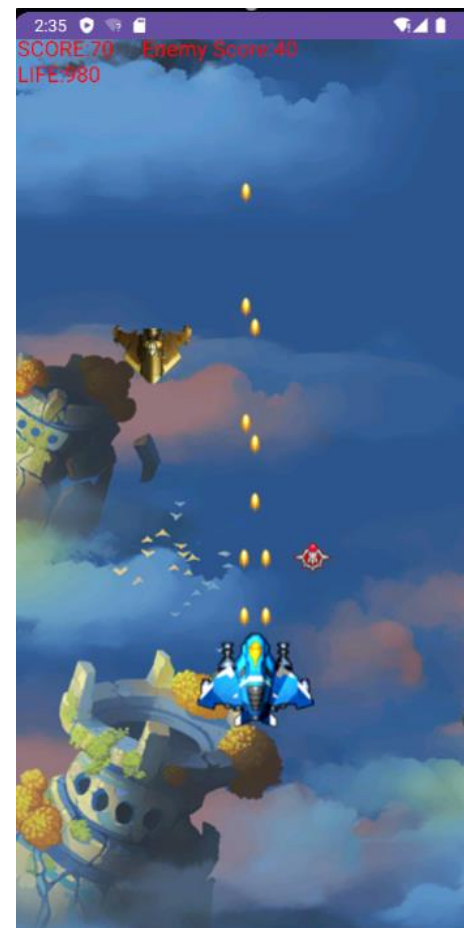
## 6-1 Socket客户端

1. 在AndroidManifest.xml中添加网络访问权限;
2. 在activity\_main.xml中添加“联机对战”按钮;
3. 监听“联机对战”点击事件, 点击后**创建子线程**连接Socket服务器;
4. 使用**10.0.2.2和9999**端口创建Socket对象, 连接到Socket服务器等待匹配对手, 等待过程使用AlertDialog显示“匹配中, 请等待.....”。



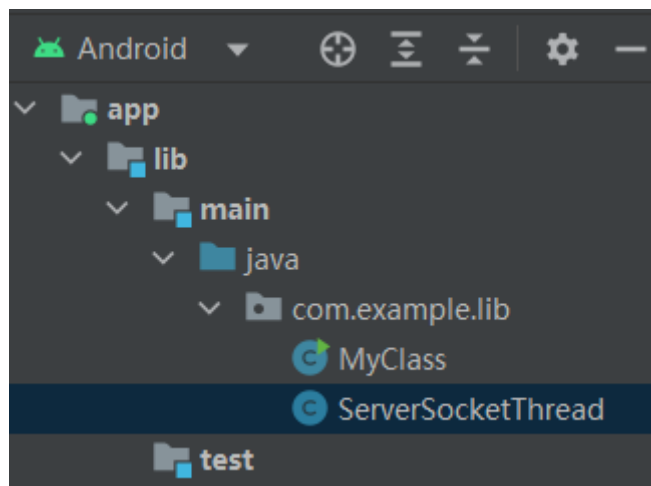
## 6-1 Socket客户端

6. 匹配成功后进入游戏主界面，实时显示对手分数；
7. 若一方英雄机死亡但对手英雄机仍存活，应继续更新对手分数，直至对手英雄机死亡后游戏结束；
8. 对战双方英雄机均死亡后断开Socket连接，显示对战结果页面，双方玩家对战结果应一致，对战结果可不保存。



## 6-2 Socket服务器端

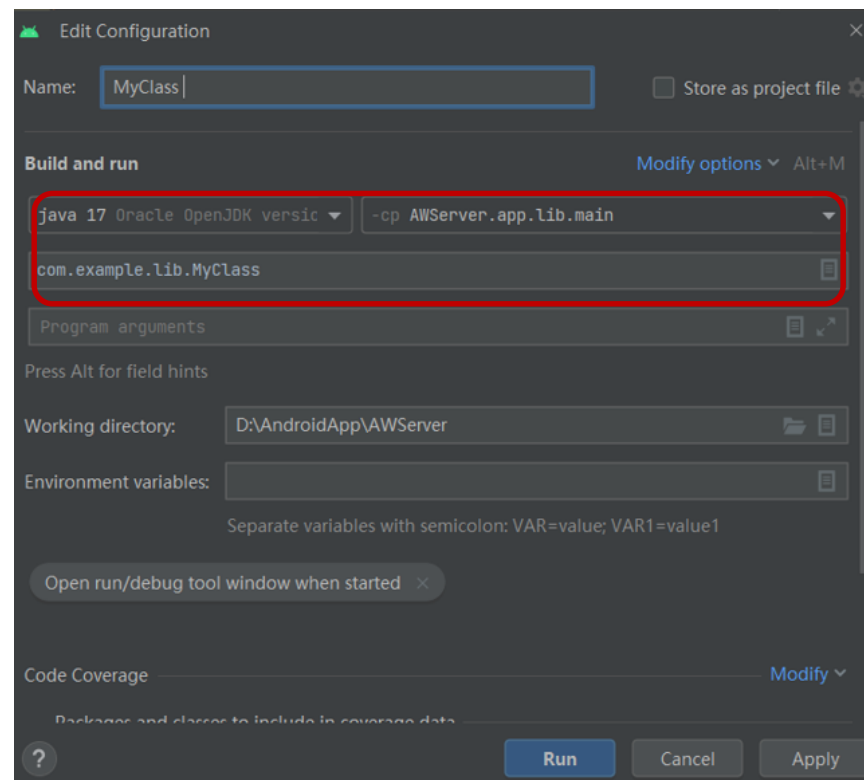
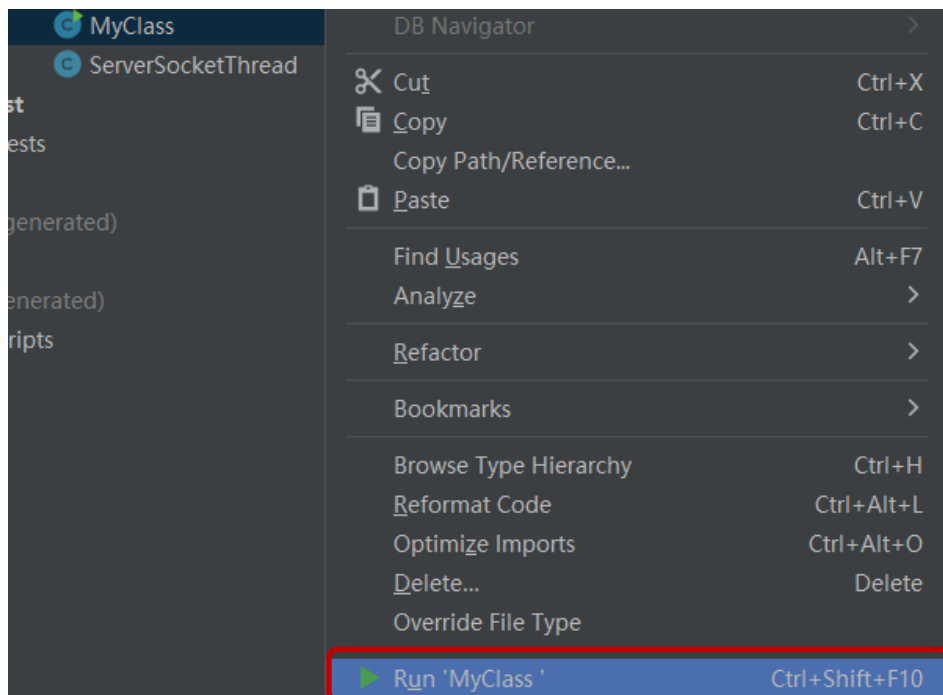
1. 解压缩AWServer项目到目标文件夹，使用Android Studio打开解压缩后的项目，在项目的app->lib模块下，MyClass和ServerSocketThread是Socket服务器端的代码框架；



2. 编写服务器端代码实现玩家的匹配和分数同步。

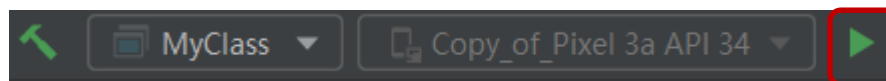
## 测试步骤

1. 测试时，先启动服务器端，右键单击MyClass，选择“ Run MyClass” ，在打开的Edit Configuration中进行如下配置；

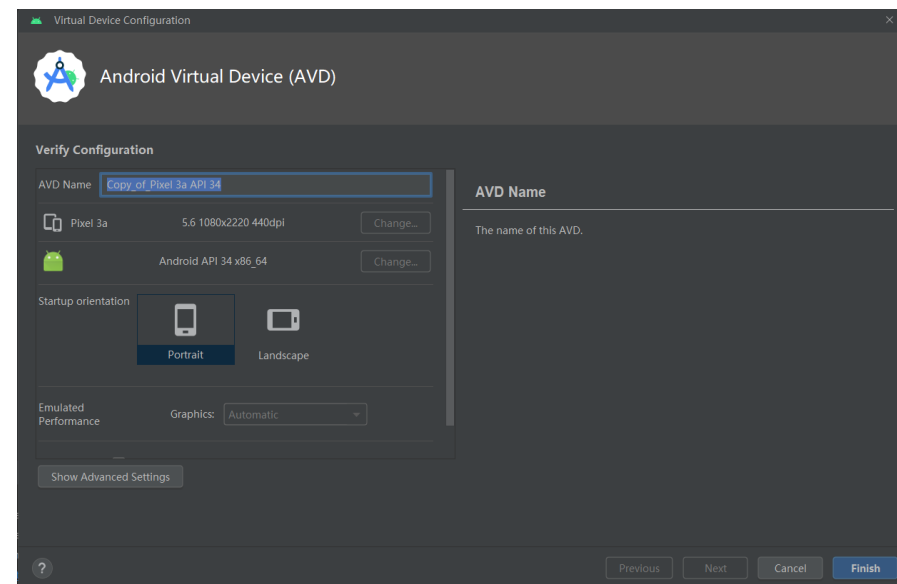
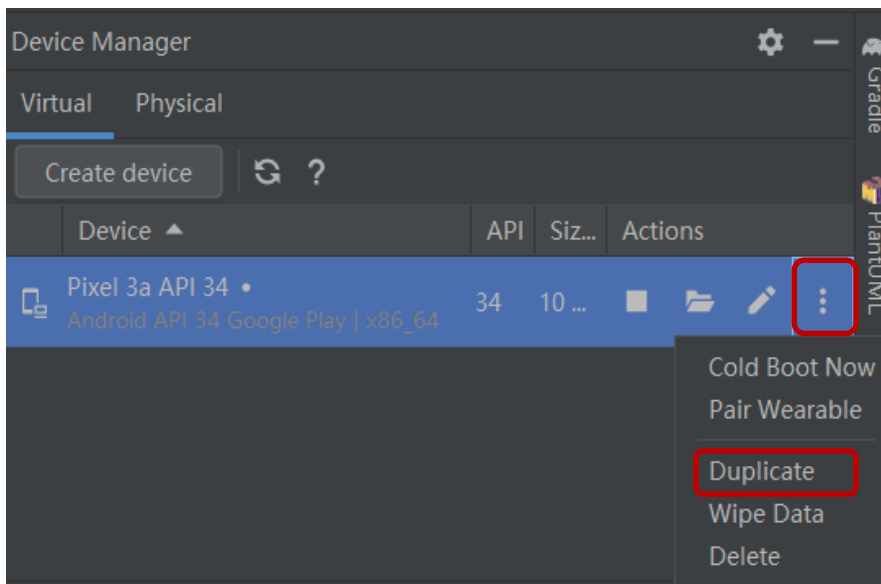


## 测试步骤

2. 点击工具栏上的Run即可运行服务器端项目；



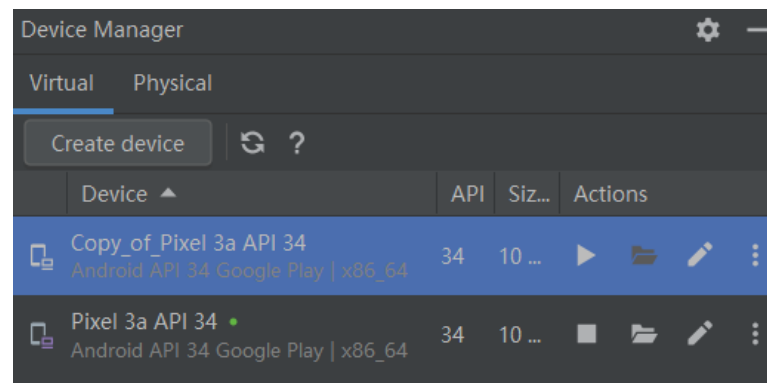
3. 在Android Device Manager面板中，选择复制当前Device，在弹出的对话框中直接选择Finish；



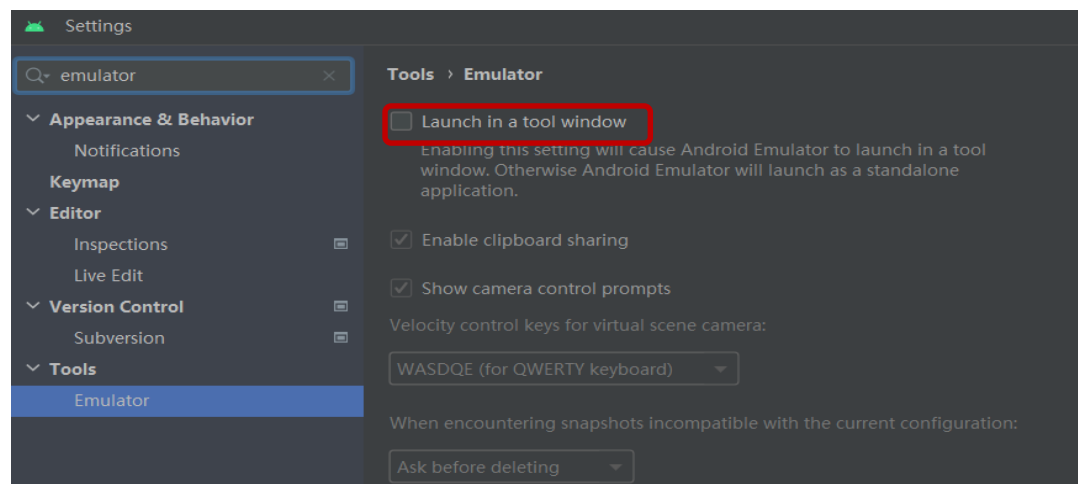


## 测试步骤

4. 复制成功后，Android Device Manager中有两个相同配置的模拟器（也可以创建不同的模拟器，注意模拟器选择**API Level 34**）



5. 在Settings->Emulator中取消勾选“Launch in a tool window”，模拟器将作为独立的应用程序打开，便于测试；



## 测试步骤

6. 依次选择两个模拟器运行，测试对战功能。

